AMENDMENTS TO THE SPECIFICATION

Please replace paragraph [0050] with the following amended paragraph.

[0050] The content publisher generates the sharing polynomial *f(x)* ~~over a finite field Z_N~~ where $a_o = SK$. Although polynomial interpolation is described, other collections of functions may also be utilized. Each partial secret share S_i may then be calculated using Equation (3), which is shown as follows:

$$S_i = f(id_i) \bmod \cancel{N}\,\underline{\phi(N)}, \qquad (3)$$

. where N is a RSA modulus and $\phi(N)$ is a Euler totient function.

Please replace paragraph [0053] with the following amended paragraph.

[0053] At block 514, for instance, the content publisher may broadcast k public witnesses of the sharing polynomial's coefficients, which are denoted as $\{g^{a_0},\cdots;g^{a_{k-1}}\}$, where ~~$g \in Z_N$~~ $\underline{g \in Z_N^*}$. After broadcast, the content publisher may destroy the polynomial. At block 516, each license authority id_i verifies validity of the received partial secret share. Validity may be checked by determining if Equation (4), as shown below, holds for the received partial secret share S_i utilizing the sharing polynomial's coefficients which were broadcast at block 514:

$$g^{S_i} = g^{a_0} \cdot (g^{a_1})^{id_i} \cdot \ldots \cdot (g^{a_{k-1}})^{id_i^{k-1}} \ \underline{\bmod N} \qquad (4)$$

In this way, each license authority id_i, may verify the validity of the received partial secret share S_i without exposing or knowing the secret, i.e. the private key SK.

Please replace paragraph [0063] with the following amended paragraph.

[0063] At block 620, the content player, when executed by the client device, determines if k correct partial licenses have been received by validating each of the partial licenses. The partial licenses may be validated as follows. First, node p calculates

$$g^{S_i} = g^{a_0} \cdot (g^{a_1})^{id_i} \cdot \ldots \cdot (g^{a_{k-1}})^{id_i^{k-1}} \ \underline{\bmod N} \qquad (7)$$

from the public witnesses of the sharing polynomial's coefficients, as was described in relation to block 516 of FIG. 5 and Equation (4). Equation (6) is then applied to g^{S_i} and the received partial license $prel_i$, A_1, and A_2 to calculate c. The received partial license $prel_i$ is verified by checking if the following equations hold: $g^r \cdot (g^{S_i})^c = A_1$ and $prel^r \cdot (prel_i)^c = A_2$. The above steps are repeated until the node p obtains k valid partial licenses. If k valid partial licenses cannot be obtained, generation of the formal-license fails (block 622).

Please replace paragraph [0064] with the following amended paragraph.

[0064] If k valid partial licenses are obtained, then at block 624, the content player combines the partial licenses to form the formal license. For example, the node p uses the k valid partial results to calculate the formal license utilizing Equation (8):

$$license = \prod_i (prel_i)^{l_{id_i}(0)} = (prel)^{\sum_i S_i \cdot l_{id_i}(0)} = (prel)^{SK} = ((license)^{PK})^{SK} \underline{\bmod N}, \qquad (8)$$

where $l_{id_i}(x) = \prod_{j=1, j \neq i}^{k} \frac{x - id_j}{id_i - id_j}$.

Please replace paragraph [0075] with the following amended paragraph.

[0075] At periodic intervals, for example, the license authorities may update their respective shares of the private key *SK* through execution of the respective update module 222 of FIG. 2. At block 802, each license authority *i* generates a random *(k, m)* sharing of the secret *0* using a random update polynomial $f_{i,\ update}(x)$, as shown in Equation (9):

$f_{i,update}(x) = b_{i,1}x + \ldots + b_{i,k-1}x^{k-1}$ ~~mod N~~ (9)